Medical Privacy and Business Process Design

John C Mitchell
Stanford

Motivating examples

- Vanderbilt Hospital Patient Portal
  - Messaging system that route requests, responses
  - Workflow: patient request, nurse, doctor, lab, ...
  - Privacy: compliance with HIPAA, hospital policy
- Call center, business process outsourcing
  - Scenarios
    - Bank call center – change address, check balance, ...
    - Credit charge disputes – receipt of goods, complaints
  - Worker does a step in task, generates new steps
  - Privacy issues: what customer data is seen, used?

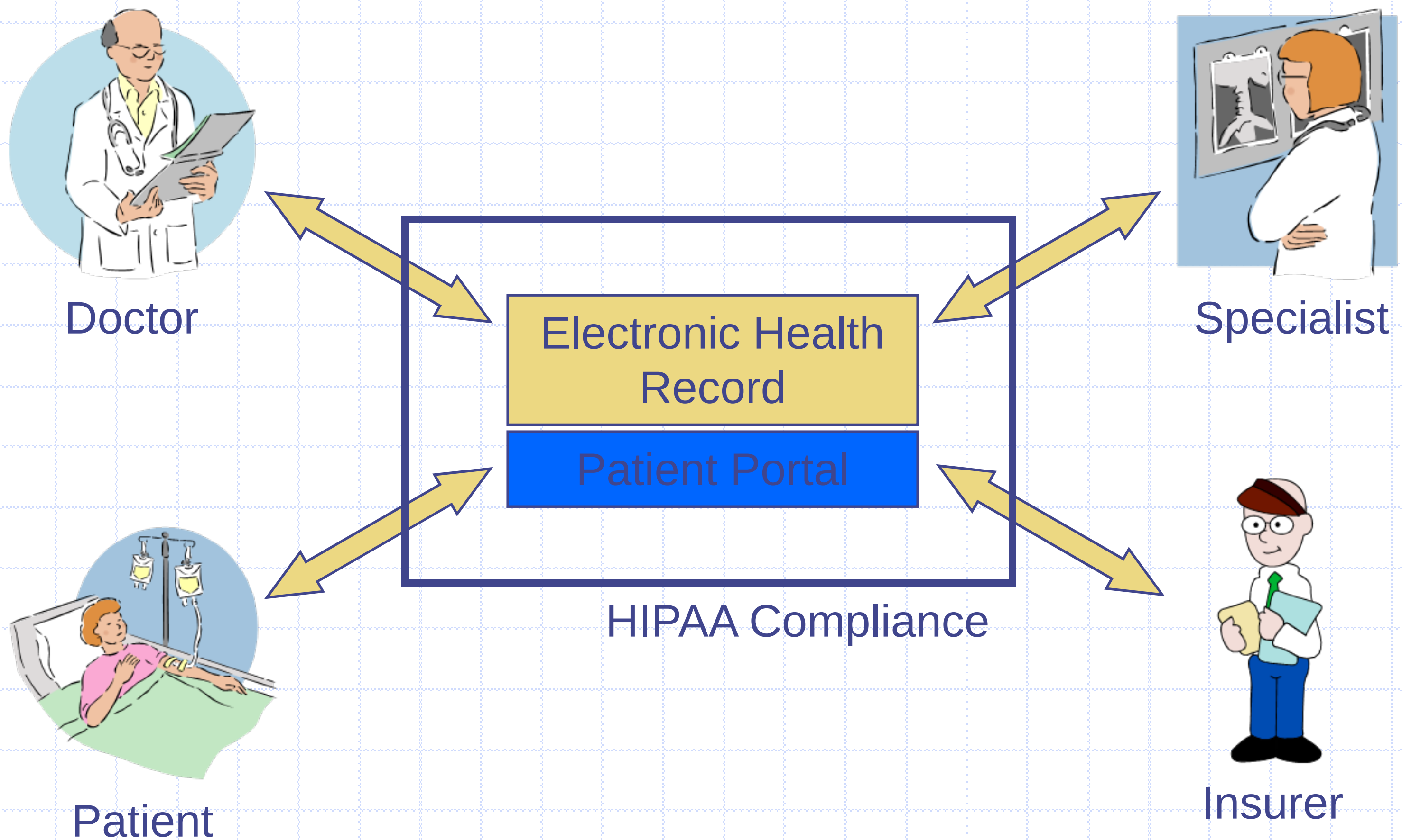
This talk

- Focus on privacy
  - Important issue in healthcare, financial services
  - Business risk – lost CCN means lost $$$
  - Regulatory compliance
    - Many organizations are uncertain what they must do to comply, not sure *how* to either
- Discovered larger set of problems
  - Need-to-know depends on step in task at hand
  - Can design business process to minimize data exposure

What is privacy?

- Intuition
  - Alice can choose who sees information about her
- Reality
  - Some kinds of information are public
  - Privacy is about “sensitive” information
    - Sensitive information *is* available to *some* by convention
      - Your bank knows your credit card number
      - Your doctor can see your medical records
    - Privacy breach occurs if sensitive information is seen or used *in violation of accepted conventions*

Example: Privacy in Health Care



Each party is conventionally allowed a different view of data

Why is privacy important

- Individuals expect privacy
  - Bank that leaks list of customers with over $1 million balance will lose those customers
- Regulations may require privacy
  - Healthcare, Financial services, …
- Reduce business risk
  - Limit fraud, identity theft, financial loss

Goals

- Express policy precisely
  - Enterprise privacy policies
  - Privacy provisions from legislation
- Analyze, enforce privacy policies
  - Does action comply with policy?
  - Does policy enforce the law?
- Support audit
  - Privacy breach may occur. Find out how it happened

Full contents
Enlarge current cover
Past issues/regional covers
Subscribe

NEWS ANALYSIS

POLITICS THIS WEEK

BUSINESS THIS WEEK

OPINION

Leaders
Letters

WORLD

United States
The Americas
Asia
Middle East & Africa
Europe

Personal data

The logic of privacy

Jan 4th 2007
From *The Economist* print edition

A new way to think about computing and personal information

PEOPLE do not have secret trolleys at the supermarket, so how can it be a violation of their privacy if a grocer sells their purchasing habits to a marketing firm? If they walk around in public view, what harm can cameras recording their movements cause? A company is paying them to do a job, so why should it not read their e-mails when they are at work?

How, what and why, indeed. Yet, in all these situations, most people feel a sense of unease. The technology for gathering, storing, manipulating and sharing information has become part of the scenery, but there is little guidance on how to resolve the conflicts created by all the personal data now washing around.

A group of computer scientists at Stanford University, led by John Mitchell, has started to address the problem in a novel way. Instead of relying on rigid (and easily programmable) codes of what is and is not acceptable, Dr Mitchell and his colleagues Adam Barth and Anupam Datta have turned to a philosophical theory called contextual integrity. This theory acknowledges that people do not require complete privacy. They will happily share information with others as long as certain social norms are met. Only when these norms are contravened—for example, when your psychiatrist tells the personnel department all about your consultation—has your privacy been invaded. The team think contextual integrity can be used to express the conventions and laws surrounding privacy in the formal vernacular of a computer language.

Privacy Model: “Contextual Integrity”

- Model disclosure, use of personal information
  - Messages has sender, receiver, subjects
- Privacy depends on context, sequence of actions
  - Past and future relevant
- Agents reason about attributes
  - Deduction based on combining information

Gramm-Leach-Bliley Example

Sender role · Attribute · Subject role

Financial institutions must notify consumers if they share their non-public personal information with non-affiliated companies, but the notification may occur either before or after the information sharing occurs

Recipient role · Transmission principle

HIPAA Example

- English policy
  - Patients can access their protected health information held by covered entities, except for their psychotherapy notes (which can be accessed after a psychiatrist approves).

- Formal policy

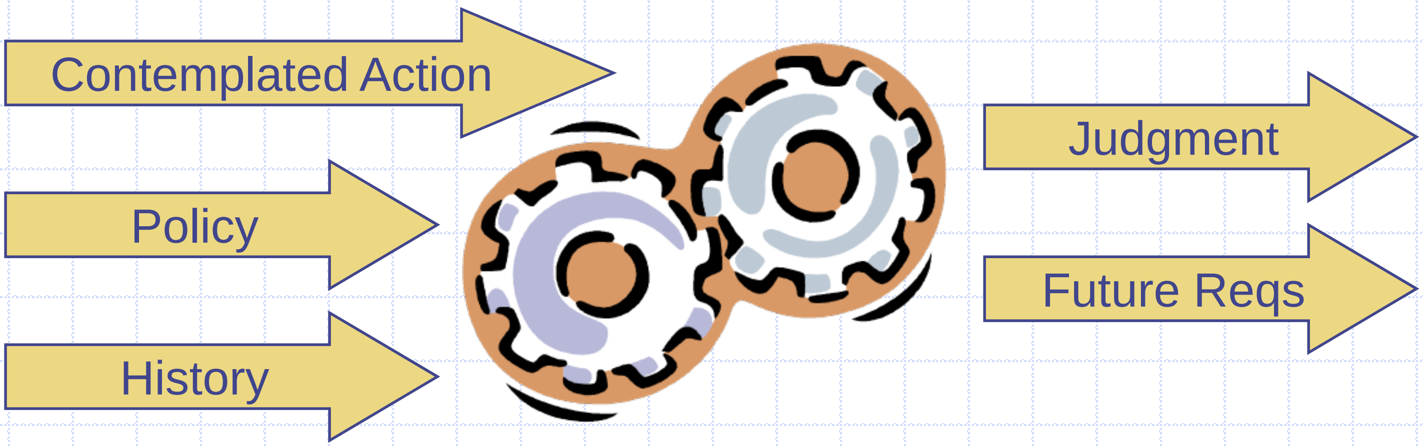
  \+ send(*p*, *q*, *m*) and inrole(*p*, *covered-entity*) and inrole(*q*, *patient*) and contains(*m*, *q*, *protected-health-information*)

  \- If send(*p*, *q*, *m*) and inrole(*p*, *covered-entity*) and inrole(*q*, *patient*) and contains(*m*, *q*, *psychotherapy-notes*), then previously send(*p′*, *p*, *m′*) and inrole(*p′*, *psychiatrist*) and contains(*m′*, *q*, *approve-disclosure-of-psychotherapy-notes*)

Refinement and Combination

- Policy refinement
  - Basic policy relation
  - Does hospital policy enforce HIPAA?
- P_1 refines P_2 if $P_1 \rightarrow P_2$
  - Requires careful handling of attribute inheritance
- Combination becomes logical conjunction
  - Defined in terms of refinement

Compliance



- Strong compliance
  - Future requirements after action can be met
  - Theorem: decidable in PSPACE
- Weak compliance
  - Present requirements met by action
  - Theorem: decidable in Polynomial time

What problem does CI solve?

- Can formulate set of allowed uses and transmissions of information
- Can check whether sequence of actions satisfies policy

What next?

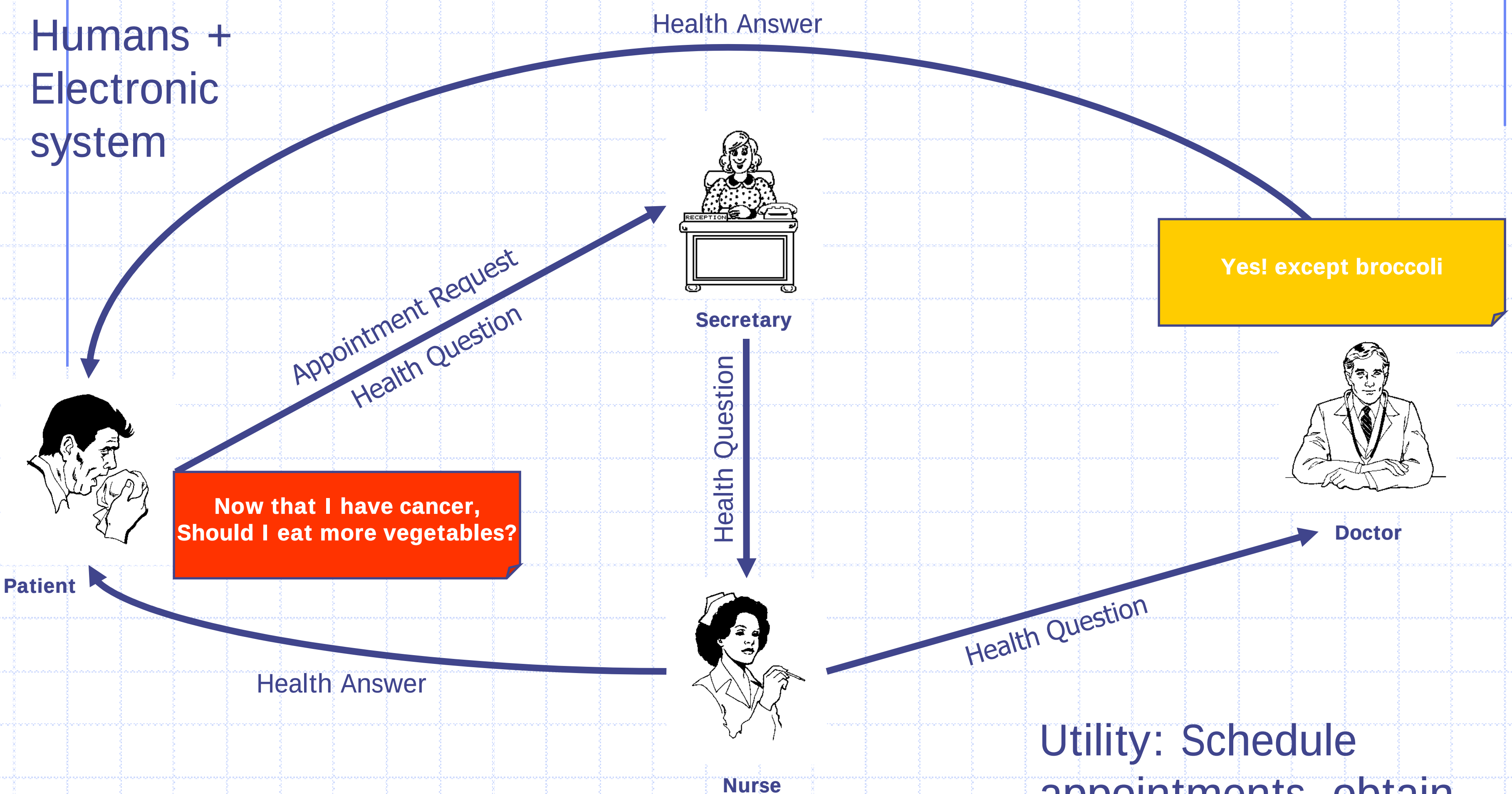
- How does an organization structure its business processes to satisfy policy?
- Some actions done by people, not computers
- What about audit, other problems?

Privacy, Utility, and Responsibility in Business Processes

Adam Barth
Anupam Datta
John Mitchell
Sharada Sundaram

MyHealth@Vanderbilt Workflow

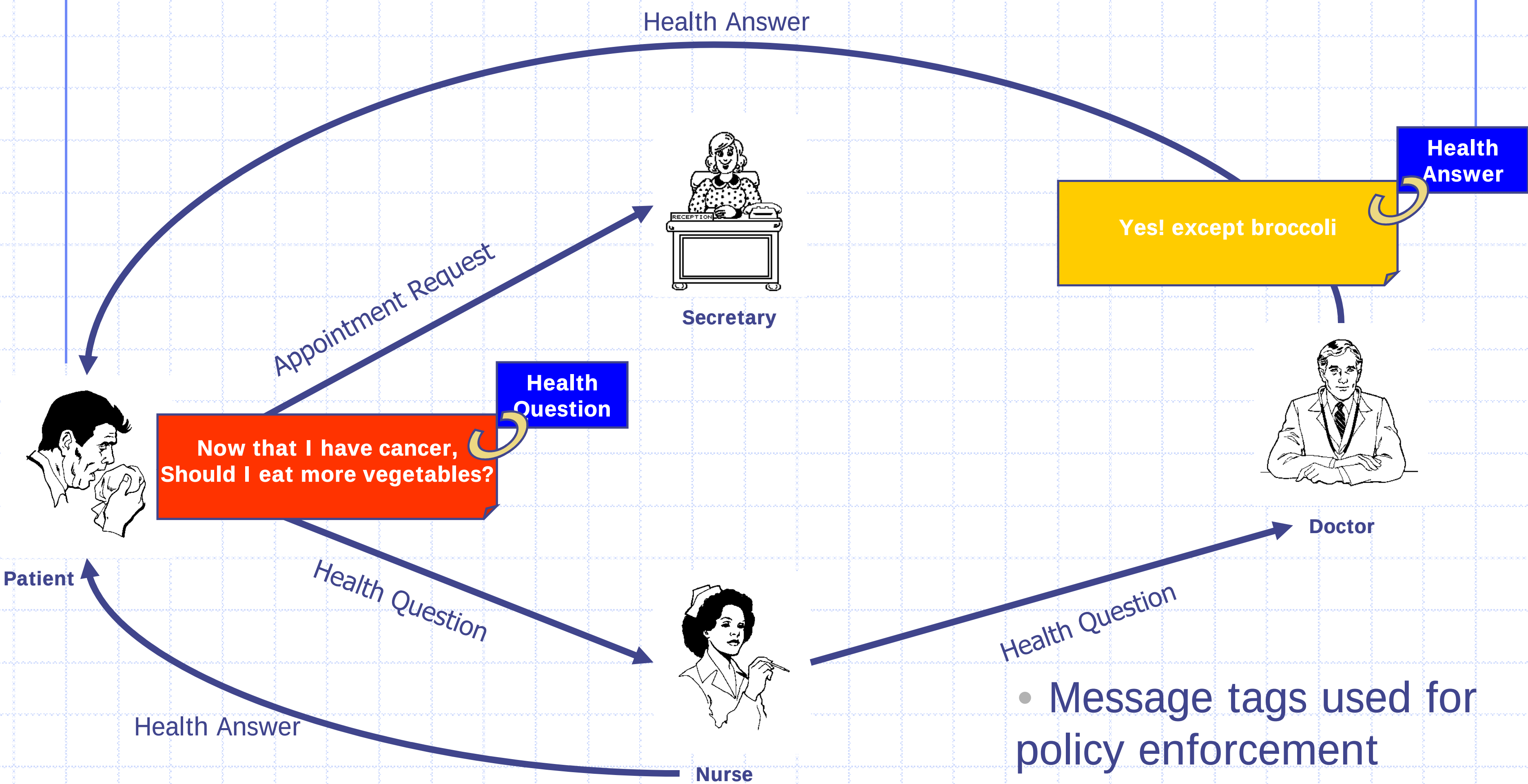
Humans + Electronic system



Utility: Schedule appointments, obtain health answers

Privacy: HIPAA compliance+

MyHealth@Vanderbilt Improved



- Message tags used for policy enforcement
- Minimal disclosure

Logic of Privacy and Utility

- Syntax

φ ::= send(p_1,p_2,m)	p_1 sends p_2 message m
| contains(m, q, t)	m contains attrib t about q
| tagged(m, q, t)	m tagged attrib t about q
| inrole(p, r)	p is active in role r
| $t \leq t'$	Attrib t is part of attrib t'
| $\varphi \wedge \varphi$ | $\neg\varphi$ | $\exists x.\ \varphi$	Classical operators
| $\varphi \mathrm{U} \varphi$ | $\varphi \mathrm{S} \varphi$ | $\mathrm{O}\varphi$	Temporal operators
| $\langle\langle \underline{p} \rangle\rangle \varphi$	Strategy quantifier

- Semantics

Formulas interpreted over concurrent game structure

Specifying Privacy

- MyHealth@Vanderbilt

  In all states, only nurses and doctors receive health questions

$G\ \forall$ p1, p2, q, m

send(p1, p2, m) $\wedge$ contains(m, q, health-question)

$\Rightarrow$ inrole(p2, nurse) $\vee$ inrole(p2, doctor)

LTL fragment can express HIPAA, GLBA, COPPA [BDMN2006]

Specifying Utility

- MyHealth@Vanderbilt

Patients have a strategy to get their health questions answered

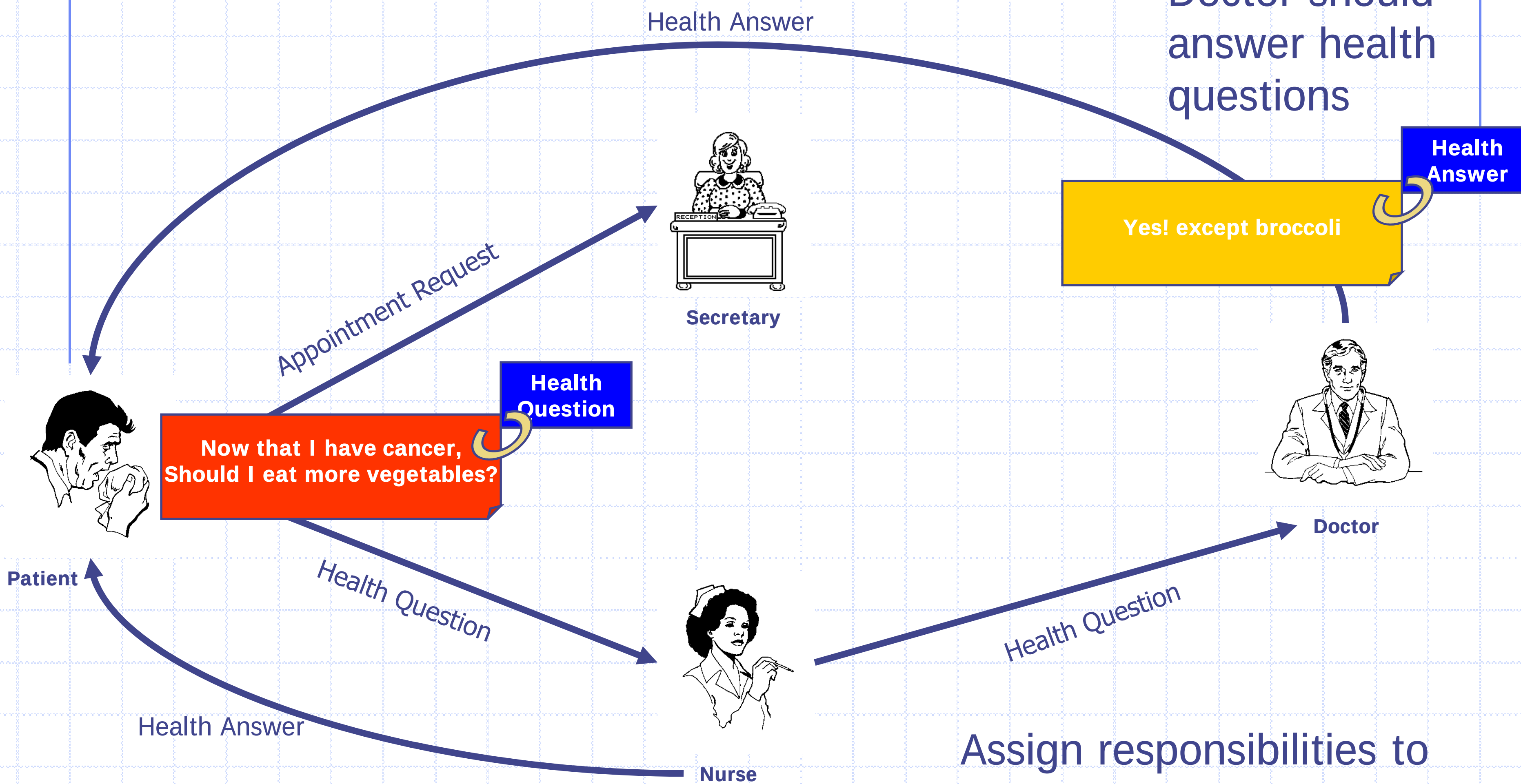
$\forall$ p inrole(p, patient) $\Rightarrow$

$\langle\langle p \rangle\rangle\ F\ \exists$ q, m.

send(q, p, m) $\wedge$ contains(m, p, health-answer)

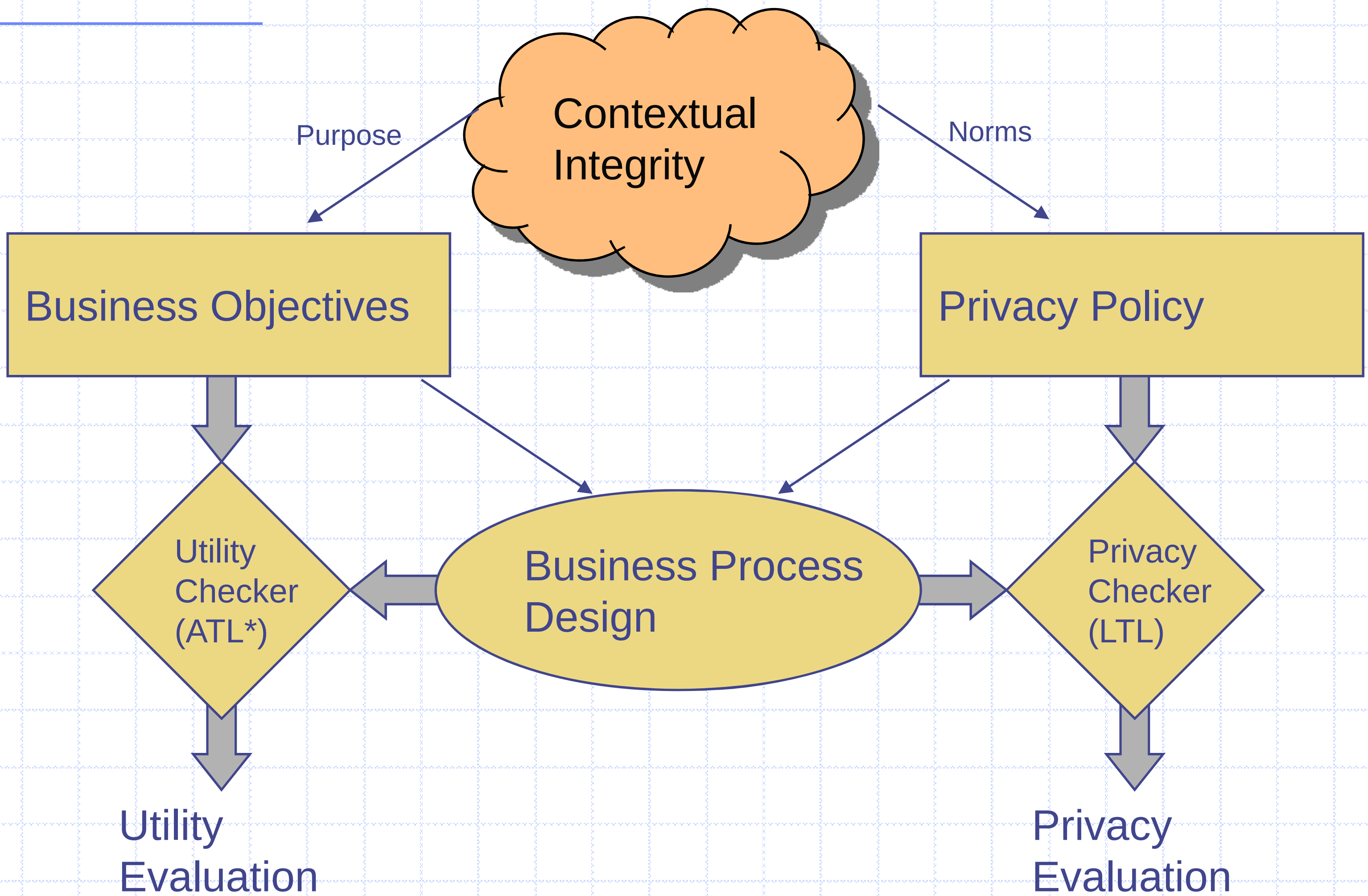
MyHealth@Vanderbilt Improved

Doctor should answer health questions



Assign responsibilities to roles & workflow engine

Design-time Analysis: Big Picture



Assuming agents *responsible*

MyHealth Responsibilities

- Tagging
  - Nurses should tag health questions

    G $\forall$p, q, s, m. inrole(p, nurse) $\wedge$ send(p, q, m) $\wedge$ contains(m, s, health-question)

    $\Rightarrow$ tagged(m, s, health-question)
- Progress
  - Doctors should answer health questions

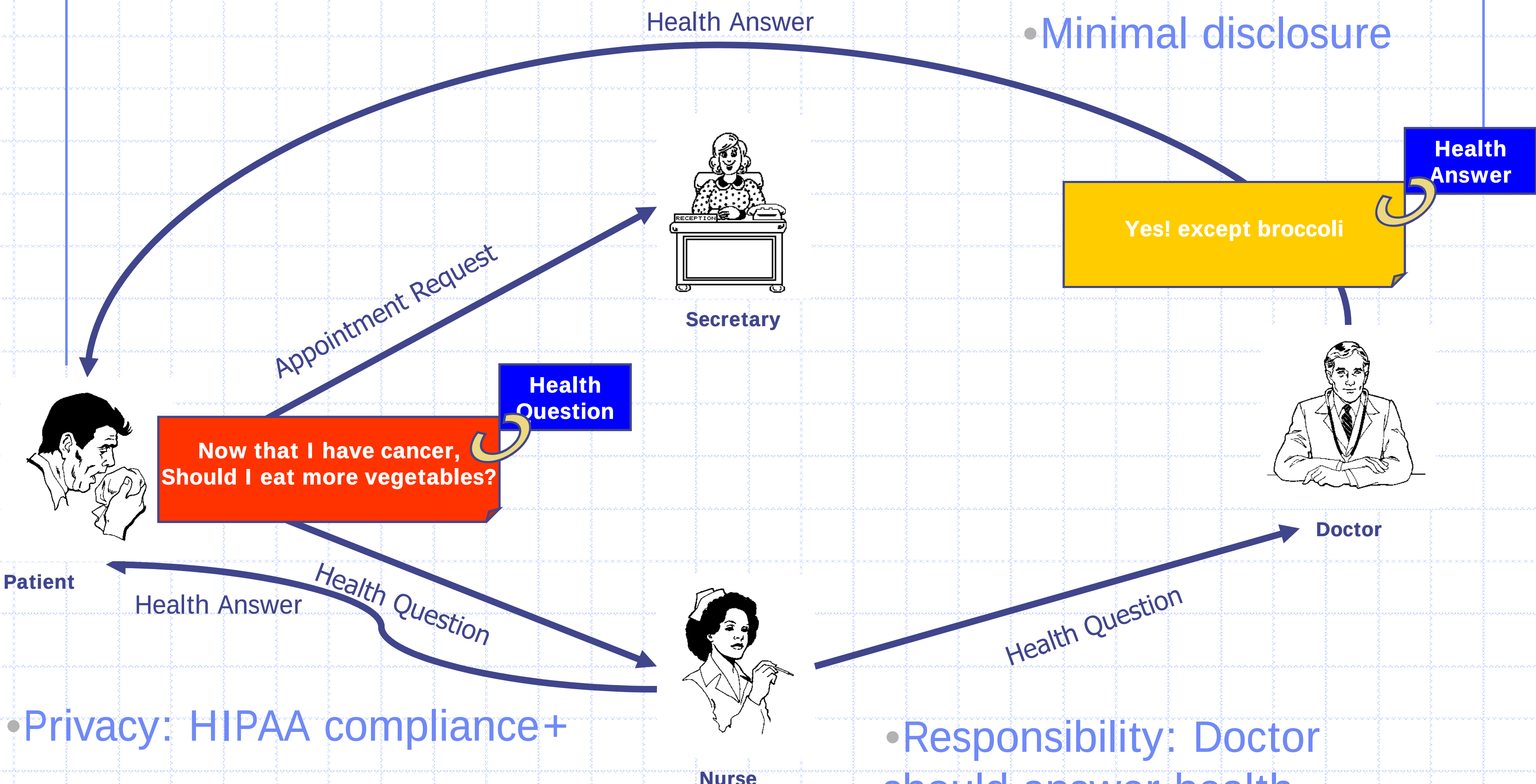
    G $\forall$p, q, s, m. inrole(p, doctor) $\wedge$ send(q, p, m) $\wedge$ contains(m, s, health-question) $\Rightarrow$

    F $\exists$m'. send(p, s, m') $\wedge$

    contains(m', s, health-answer)

MyHealth@Vanderbilt Improved

- Minimal disclosure



- Privacy: HIPAA compliance+
- Utility: Schedule appointments, obtain health answers
- Responsibility: Doctor should answer health questions

Workflow Design Results

- Theorems:

  Assuming all agents act responsibly, checking whether workflow achieves

  - Privacy is in PSPACE (in size of workflow formula)
  - Utility is decidable

- Definition and construction of minimal disclosure workflow

Algorithms implemented in model-checkers, e.g. SPIN, MOCHA

Deciding Privacy

- PLTL model-checking problem is PSPACE decidable

$\boldsymbol{G}$ |= tags-correct $\boldsymbol{U}$ agents-responsible
$\Rightarrow$ privacy-policy

$\boldsymbol{G}$: concurrent game structure

Result applies to finite models (#agents, msgs,…)

MyHealth Privacy

- MyHealth@Vanderbilt workflow satisfies this privacy condition

  In all states, only nurses and doctors receive health questions

$$G\ \forall\ p1, p2, q, m$$
$$send(p1, p2, m) \wedge contains(m, q, health\text{-}question)$$
$$\Rightarrow inrole(p2, nurse) \vee inrole(p2, doctor)$$

- Run LTL model-checker, e.g. SPIN

Deciding Utility

- ATL* model-checking of concurrent game structures is
  - Decidable with perfect information
  - Undecidable with imperfect information
- Theorem:

  There is a sound decision procedure for deciding whether workflow achieves utility
- Intuition:
  - Translate imperfect information into perfect information by considering possible actions from one player's point of view

MyHealth Utility

- MyHealth@Vanderbilt workflow satisfies this utility condition

  Patients have a strategy to get their health questions answered

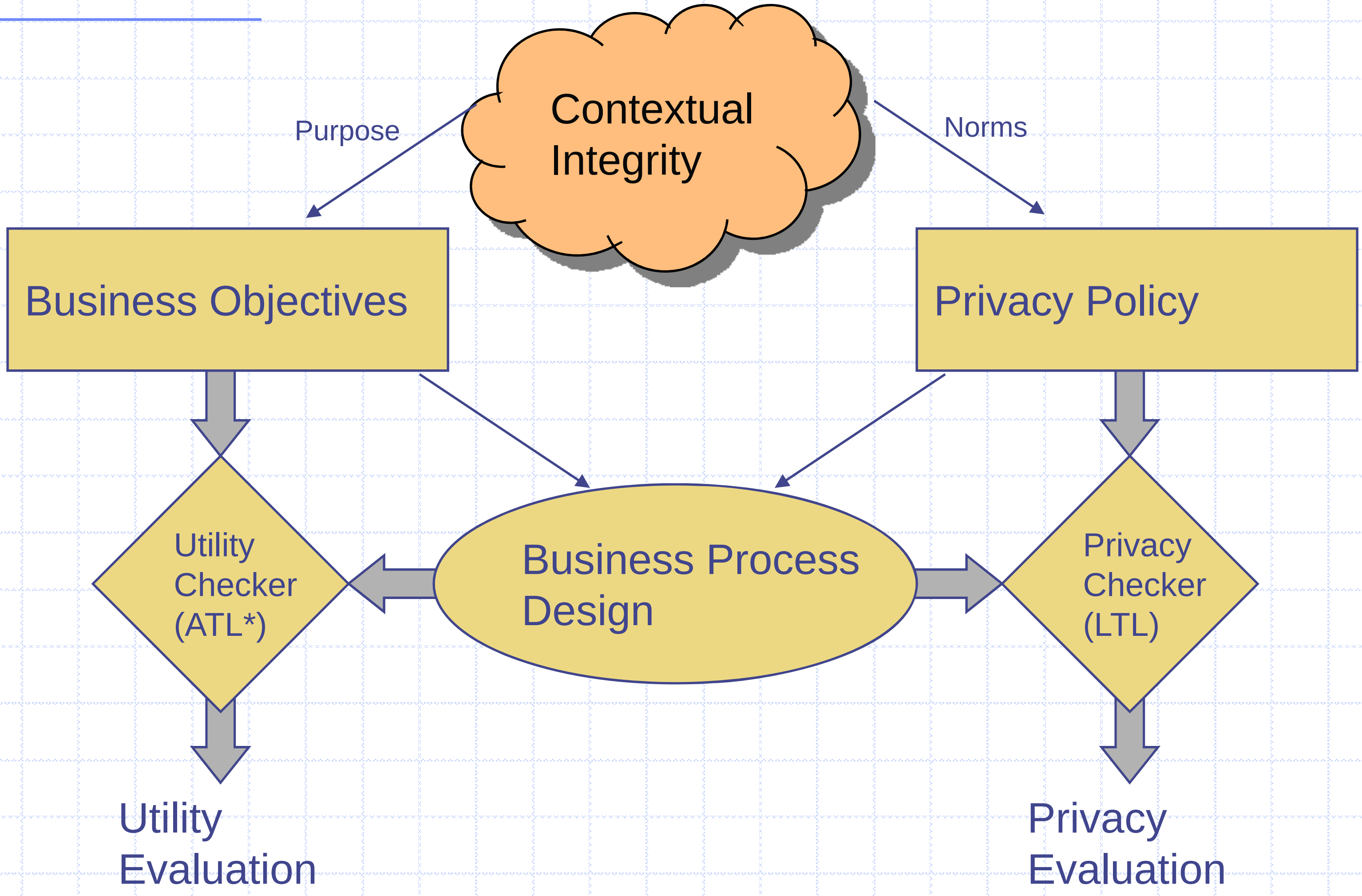
  $\forall$ p inrole(p, patient) $\Rightarrow$

  <<p>> F $\exists$ q, m.

  send(q, p, m) $\wedge$ contains(m, p, health-answer)

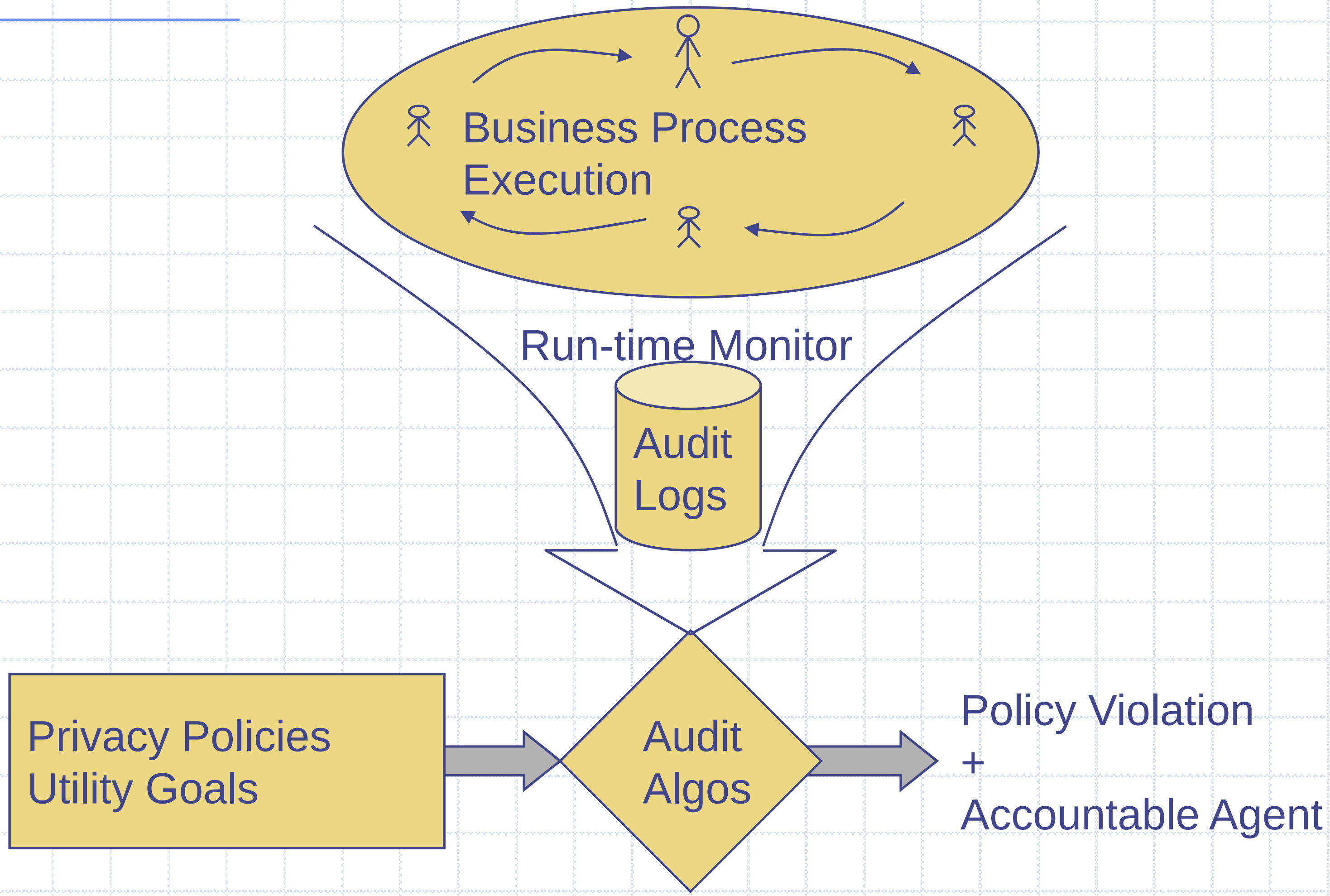
- Run ATL* model-checker, e.g. MOCHA

Design-time Analysis: Big Picture



Assuming agents *responsible*

Auditing: Big Picture



Auditing Results

- Definitions
  - Policy compliance, locally compliant
  - Causality, accountability
- Design of audit log
- Algorithms
  - Finding agents accountable for locally-compliant policy violation in graph-based workflows using audit log
  - Finding agents who act irresponsibly using audit log
- Algorithms use oracle:
  - $\mathcal{O}$(msg) = contents(msg)
  - Minimize number of oracle calls

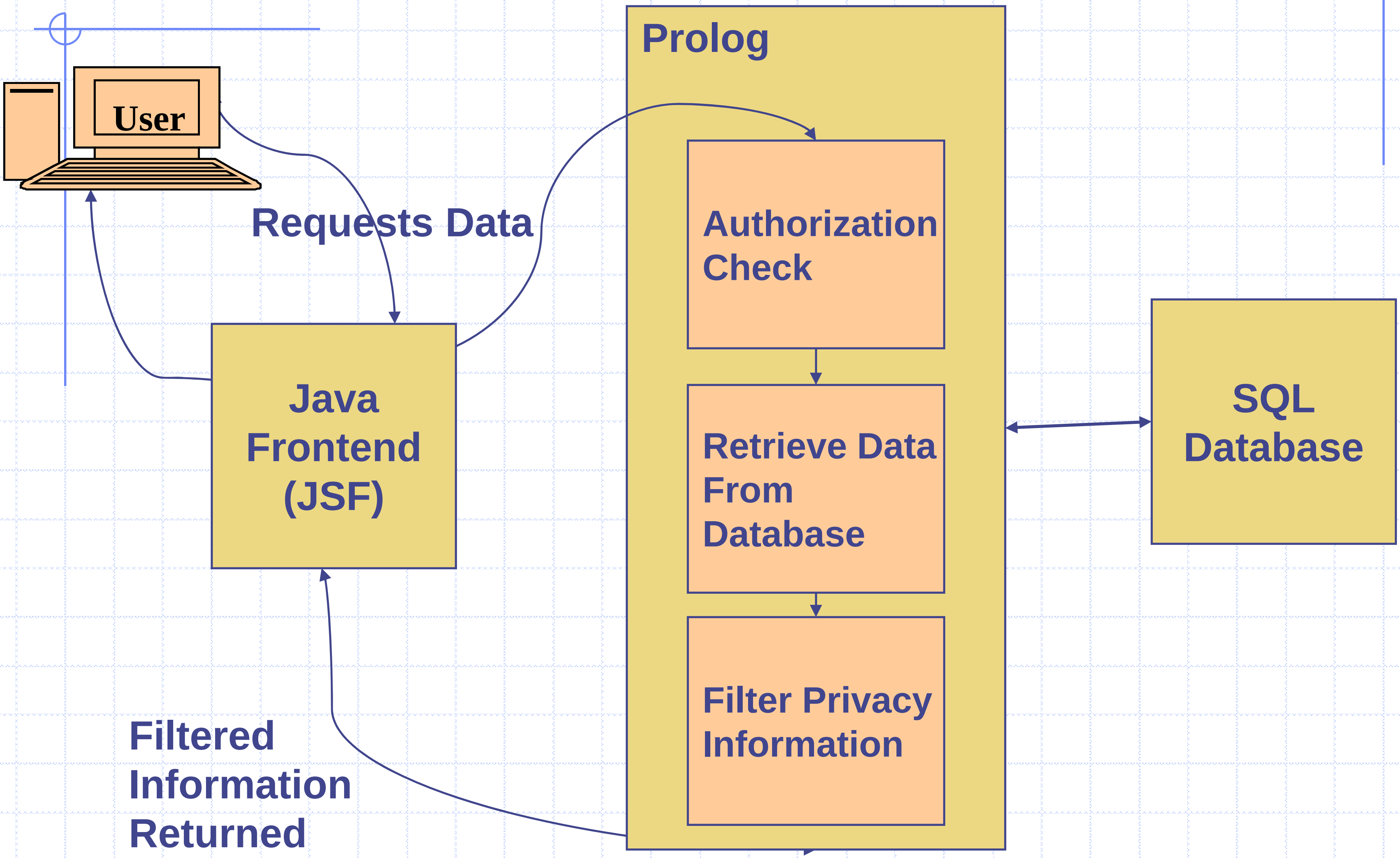
Auditing Algorithm

- Goal
  - Find agents accountable for a policy violation
- Algorithm(Audit log A, Violation v)
  - Construct G, the causality graph for v in A
  - Run BFS on G.
    - At each Send(p, q, m) node, check if tags(m) = O(m). If not, and p missed a tag, output p as accountable
- Theorem:
  - The algorithm outputs at least one accountable agent for every violation
    - of a locally compliant policy in an audit log
    - of a graph-based workflow that achieves the policy in the responsible model

Summer 2007 project

- Construct demo patient portal web site
  - Explore surrogate, delegate issues
  - Show Vanderbilt Hospital
- Use standard tool
  - JSF – Java framework for business logic
  - Prolog – XSB implementation
  - SQL Database – enterprises already store org info
- Outcome
  - Lots of time spent on mechanics of building site
  - Some insight into separating policy from UI

Information Flow



Some features we explored

- Automatic Prescriptions
- Appointment scheduling
- Asking and answering of health questions
- Delegate and Surrogate Access
- Lab and other medical information
- (Insurance view – partially completed)

Conclusions

- Framework
  - Concurrent game model
  - Logic of Privacy and Utility
    - Temporal logic (LTL, ATL*)
- Business Process as Workflow
  - Role-based responsibility for human and mechanical agents
- Algorithmic Results
  - Workflow design assuming agents responsible
    - Privacy, utility decidable (model-checking)
    - Minimal disclosure workflow constructible
  - Auditing logs when agents irresponsible
    - From policy violation to accountable agents
    - Finding irresponsible agents

} Auto-mated

} Using oracle